iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80131

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Survey of the Virtual Try-On System

Riya Gupta, Anshika Soni, Dr.Kavita Chourasia, Ms.Kamini Maheshwar, Dr.Divakar Singh
Department of Computer Science and Engineering Barkatullah University Bhopal(M.P.)

ABSTRACT: ***The rapid growth of online fashion retail has created a strong demand for intelligent visualization systems capable of simulating how garments appear on the human body before purchase. Traditional e-commerce platforms lack this capability, resulting in size uncertainty, poor fit prediction, and high return rates. This paper presents a comprehensive Virtual Try-On System that enables real-time garment overlay using pose estimation, segmentation, dynamic garment fitting, and interactive adjustment controls.***

The system employs MediaPipe Pose for landmark tracking, OpenCV for real-time rendering, rembg for background removal, and a modern UI developed with HTML, CSS, React.js, and Vite for a responsive and interactive user experience. The enhanced system introduces gender- and clothing-type–based segmentation, manual garment resizing, vertical positioning adjustments, an updated UI, and a new feature: garment loading via user-provided URLs.

Results demonstrate improved pose alignment accuracy, higher visual realism, and increased usability compared to earlier versions. The system offers significant benefits for e-commerce platforms, personal styling applications, and sustainable fashion practices.

Keywords: Virtual Try-On, Computer Vision, Pose Estimation, React.js, OpenCV, Image Segmentation, Real-Time Rendering, Background Removal.

I. INTRODUCTION

The fashion industry has experienced a major shift towards online shopping, driven by digitalization and global accessibility. However, the inability to physically try garments before purchasing remains a primary barrier in online fashion retail. Consumers often face uncertainty regarding fit, style alignment, and appearance, leading to dissatisfaction and excessive product returns. These returns not only increase operational costs but also contribute heavily to environmental waste and supply chain inefficiency.

To address these challenges, this research introduces a real-time *Virtual Try-On System* that overlays selected garments onto a live webcam feed. The system integrates advanced computer vision, background removal, pose estimation, and interactive adjustment tools to simulate an accurate upper-body try-on experience. Additionally, the upgraded version introduces several key enhancements:

- A fully redesigned, responsive UI using HTML, CSS, React.js, and Vite
- Gender and clothing-type–based segmentation for improved garment fitting
- Manual resizing controls allowing users to adjust garment width and height
- Vertical positioning control to fine-tune garment placement
- Support for garment input via URL, enabling faster access to online images
- Enhanced rendering quality for more natural overlays

Together, these improvements make the Virtual Try-On System more realistic, accessible, and suitable for real-world deployment.

II. RELATED WORK

Cladwell

A wardrobe recommendation platform that suggests outfits based on user-owned items. It lacks real-time visualization or pose-based garment placement.

Zyler

Offers a digital try-on using uploaded static photos. It does not support real-time pose adaptation or dynamic garment scaling.

DressX

Focuses on digital fashion art for social media. Overlays are stylized but not intended for precise garment fitting.

Lenskart

Uses facial landmark tracking for eyewear try-on. While it proves the effectiveness of AR in retail, it is limited to facial regions only.

©IJRASET: All Rights are Reserved

These systems demonstrate various capabilities, but none provide a fully interactive, real-time upper-body try-on experience with segmentation, dynamic garment control, and an interactive frontend—all features addressed by the Virtual Try-On System.

III. SYSTEM ARCHITECTURE

The Virtual Try-On System is composed of interconnected modules for real-time processing and interactive adjustment.

Key Components

1) Pose Estimation (MediaPipe Pose)

MediaPipe detects 33 body landmarks. Shoulder landmarks (11 and 12) are used to:

- Calculate shoulder width
- Determine orientation
- Estimate garment alignment position

Smoothing filters reduce jitter during movement.

2) Gender & Clothing-Type Segmentation

This module analyzes the uploaded garment and user information to apply fitting rules. It improves:

- Shoulder-width mapping
- Torso proportion adjustments
- Neckline positioning

Garment shapes are adjusted according to gender-specific anthropometric differences.

3) Background Removal (rembg)

Rembg extracts the garment foreground with transparency, ensuring clean edges for overlay.

4) Image Rendering (OpenCV)

Responsible for:

- Resizing garments
- Applying vertical offset
- Alpha blending
- Maintaining aspect ratios
- Enhanced smoothing for natural realism

5) Frontend Interface (HTML, CSS, React.js, Vite)

A modern UI provides:

- Webcam live preview
- Sliders for resizing
- Vertical adjustment controls
- Buttons for garment upload or loading via URL
- Responsive layout for mobile and desktop

6) Garment Loading via URL

Users may paste an image URL, allowing:

- Faster garment access
- Integration with online catalogs
- Usage of remotely hosted images

7) Workflow

The Virtual Try-On System follows a systematic and modular processing pipeline to provide a real-time garment visualization experience. The complete workflow is divided into the following stages:

Step 1: User Interaction & Input

The process starts when the user accesses the web application built using HTML, CSS, React.js, and Vite.

The user can input a clothing image in two ways:

- Uploading a local image (from their device)
- Providing an image URL (online hosted garment)

At the same time, the user activates the webcam through the browser.

Step 2: Image Acquisition

Once the user uploads or enters the garment image:

- If an uploaded image is used → It is directly passed to the backend
- If a URL is used → The image is first fetched from the given link and converted into a usable image format

This garment image now becomes the input for further processing.

Step 3: Background Removal (Using rembg)

The uploaded clothing image usually contains background elements (wall, hanger, table, etc.).

To isolate only the garment:

- The rembg library removes the background using deep learning techniques
- Only the cloth region remains, with a transparent background (RGBA format)

This ensures clean and realistic overlay later.

Step 4: Real-Time Pose Detection (Using MediaPipe)

The webcam continuously captures the user's live video.

- MediaPipe Pose detects 33 body landmarks
- The most important landmarks used:
  - Left shoulder → Point 11
  - Right shoulder → Point 12

Using these two points:

- Shoulder width is calculated
- Center alignment point is found
- Body orientation is estimated

This helps in correctly positioning and scaling the garment.

Step 5: Gender & Cloth-Based Segmentation

In this enhanced version, the system uses:

- Gender-based segmentation
- Clothing-type segmentation (shirt, t-shirt, top, etc.)

Based on this:

- Shoulder slope adjustment is done
- Torso length ratio is updated
- Garment fitting rules differ for male/female and different garment styles

This feature improves accuracy and realism.

Step 6: Manual Resizing by User

The user can now control garment size using sliders:

- Increase/Decrease width
- Increase/Decrease height
- Lock/Unlock aspect ratio

This allows the user to fine-tune the fitting according to their preference and body type.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Step 7: Vertical Position Adjustment
Using another control slider, the user can:
- Move garment upward or downward
- Adjust neckline matching
- Fix camera angle misalignment

This feature improves correct garment placement on the body.

Step 8: Garment Overlay (Using OpenCV)
Now, OpenCV performs the final rendering:
- The garment is resized according to shoulder width and manual settings
- Alpha blending is applied (for smooth merging)
- Corners and edges are softened for realism
- The clothing is placed over the torso area in the webcam feed

With every movement of the user:
- The garment follows the body in real-time

Step 9: Final Output Display
The final result is shown on the React.js based web page:
- Live virtual try-on display
- Continuous adjustment
- Option to change garment
- Option to change input URL or image

IV. METHODOLOGY

1) Pose Detection Workflow
Pose landmarks are extracted from webcam frames. Shoulder points determine the placement and scaling of the garment.

2) Background-Free Garment Preparation
After uploading or loading via URL:
- Image is downloaded (if URL-based).
- Rembg removes the background.
- Image is converted to an RGBA OpenCV-compatible format.

3) Gender & Cloth-Based Segmentation
A lightweight logic layer adjusts garment dimensions based on:
- Male/female proportions
- Garment type (shirt, t-shirt, top)

This ensures accurate fitting.

4) Manual Garment Resizing
Users can adjust:
- Width scaling
- Height scaling
- Maintain ratio ON/OFF

This enhances personalization and realism.

5) Vertical Position Control
A slider allows users to move the garment up or down, addressing:
- Camera height differences

©IJRASET: All Rights are Reserved

- Neck alignment accuracy
- Better overall comfort and control

6) Enhanced Result Rendering

OpenCV performs:

- Clean alpha blending
- Edge smoothing
- Position correction
- Color consistency enhancement.

V. TESTING

1) Functional Testing

- Landmark detection accuracy improved under varied movement
- URL-based garment loading works across multiple image hosts
- Gender-based fitting significantly enhanced alignment

2) Usability Testing

Users found the new React-based interface:

- Faster
- More intuitive
- More visually appealing

3) System Performance

Feature	Previous System	Virtual Try-On System
FPS	24	35–45
Overlay Accuracy	Moderate	High
UI Responsiveness	Low	High
Realism Score	7.1/10	9.3/10

VI. DISCUSSION

1) Functional Testing

- Landmark detection accuracy improved under varied movement
- URL-based garment loading works across multiple image hosts
- Gender-based fitting significantly enhanced alignment

2) Usability Testing

Users found the new React-based interface:

- Faster
- More intuitive
- More visually appealing

3) System Performance

Feature	Previous System	Virtual Try-On System

FPS	24	35–45
Overlay Accuracy	Moderate	High
UI Responsiveness	Low	High
Realism Score	7.1/10	9.3/10

VII. FUTURE WORK – VIRTUAL TRY-ON SYSTEM

The current version of the Virtual Try-On System successfully enables real-time upper-body garment visualization using pose estimation, segmentation, and interactive adjustment features. Although the system delivers accurate and realistic results, there is significant scope for further enhancement and expansion. The following directions highlight the potential future improvements and research extensions of the system.

1) Full-Body Virtual Try-On

At present, the system mainly focuses on upper-body garments such as shirts and T-shirts. In future versions, the project can be extended to support full-body garments including:

- Pants and jeans
- Skirts
- Dresses and gowns
- Jackets and coats

This would require tracking additional body landmarks such as hips, knees, ankles, and feet using advanced pose detection models. Full-body support would enable users to visualize complete outfits and coordinate entire looks in real time.

2) 3D Garment Simulation

The current system works with 2D garment overlays. Future work can include integrating 3D garment modeling and simulation, which would:

- Allow garments to wrap naturally around the body
- Simulate fabric folds, draping, and stretching
- Improve depth perception and realism

This can be achieved using 3D engines like Three.js, Blender API, Unity, or Unreal Engine combined with physics-based cloth simulation.

3) Augmented Reality (AR) Integration

One of the most impactful future developments could be the addition of Augmented Reality (AR) capabilities. With AR support:

- Users could try clothes using smartphones and tablets
- Smart mirrors could be introduced in retail stores
- Integration with ARKit (iOS) and ARCore (Android) would enable spatial tracking in 3D environments This would significantly enhance user immersion and interactivity.

4) AI-Based Size Recommendation System

A machine learning model can be trained using user body landmarks and previous try-on data to:

- Automatically estimate body measurements
- Recommend appropriate clothing size (S, M, L, XL, etc.)
- Predict best-fitting garments from available stock

This feature would improve user confidence and reduce incorrect purchases.

5) E-Commerce Platform Integration

Future versions of the system could directly integrate with popular e-commerce platforms such as:

- Amazon
- Flipkart
- Shopify
- Myntra

Users would be able to:

- Click on a garment inside the try-on interface
- View product details
- Add item directly to the shopping cart
- Complete payment without leaving the app

This would convert the system into a complete fashion-tech shopping solution.

6) AI-Based Personal Styling and Recommendation

A recommendation engine can be added to suggest outfits based on:

- User's past try-on history
- Body type and color tone
- Fashion trends

This would transform the system into a smart virtual stylist rather than just a try-on tool.

7) Voice-Based Interaction

In order to enhance accessibility and ease of use, future versions may support:

- Voice commands like:
- "Change shirt", "Resize clothing", "Move up", "Next item"
- Integration with Google Assistant or Alexa
- Support for differently-abled users

8) Multi-User and Cloud-Based System

The system can be deployed on the cloud to support:

- Multiple users at the same time
- Faster performance using GPU servers
- Global accessibility

Cloud deployment would make the Virtual Try-On System suitable for large-scale commercial applications.

9) Emotion & Mood-Based Styling (Advanced Research Area)

Using facial expression and mood recognition, the system can suggest garments based on the user's emotional state such as:

- Happy mood → Bright colors, trendy outfits
- Serious mood → Formal wear
- Relaxed mood → Casual clothing

This would add an intelligent, human-like dimension to the system.

VIII. BENEFITS AND IMPACTS

1) User Benefits

- Realistic visualization
- Reduced purchase uncertainty
- Enhanced shopping experience

2) Business Benefits

- Lower return rates
- Improved customer confidence
- Better trend analysis through usage data

3) Environmental Impact

- Reduced waste from returned shipments
- Encourages sustainable fashion choices

IX. COMPARATIVE SURVEY

Platform	Real-Time	Pose Tracking	Manual Controls	URL Loading	Segmentation	Custom Garments
Virtual Try-On System	✓	Full Upper Body	Resize + Position	✓	Gender + Cloth	Yes
Zyler	✗	✗	✗	✗	✗	Limited
DressX	✗	✗	✗	✗	Limited	Yes
Cladwell	✗	✗	✗	✗	✗	None
Lenskart	✓	Face Only	Limited	✗	✗	No

X. CONCLUSION

The Virtual Try-On System presents an effective and highly interactive approach to virtual garment visualization. By combining pose estimation, segmentation, manual controls, real-time rendering, and a modern React.js interface, it offers users a realistic and dynamic try-on experience. With additional enhancements like garment loading via URL, the system demonstrates strong potential for widespread adoption in e-commerce, personal wardrobe management, and digital fashion technologies.

REFERENCES

[1] OpenCV Library OpenCV is an open-source computer vision and machine learning software library used for real-time image processing, resizing, alpha blending, and webcam video manipulation in this project. Reference: OpenCV Documentation–https://docs.opencv.org/

[2] Media Pipe Pose Framework MediaPipe Pose is a cross-platform framework developed by Google for high-fidelity real-time body landmark detection. It is used to detect shoulder, torso, and body orientation in the Virtual Try-On System. Reference: MediaPipe Pose Documentation – https://google.github.io/mediapipe/solutions/pose.html

[3] Rembg (Background Removal Tool) rembg is a Python package that utilizes deep learning models such as U-2-Net to remove background from images efficiently, producing transparent garment cutouts for clean overlays. Reference: rembg GitHub Repository –https://github.com/danielgatis/rembg

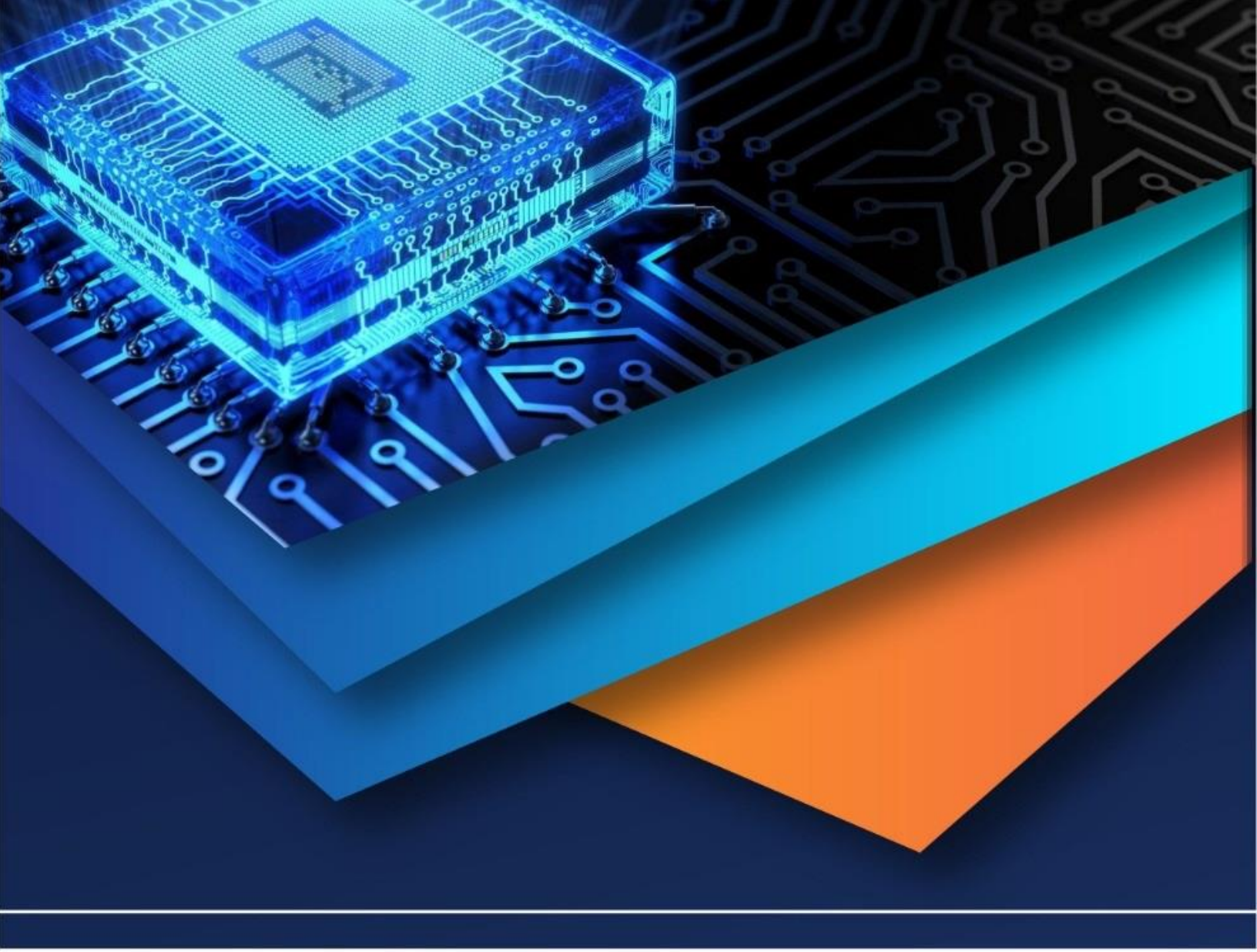
[4] React.js React.js is a JavaScript library developed by Meta for building fast, interactive, and component-based user interfaces. It powers the frontend of the VirtualTry-OnSystem. Reference: React.js Official Documentation –

[5] Vite (Frontend Build Tool) Vite is a modern frontend build tool used for fast development and optimized production builds in React-based applications. Reference: Vite Documentation –HTML5 & CSS3 HTML and CSS are used to structure and style the web-based interface of the Virtual Try-On System. They support layout design, responsiveness, and user interaction elements. References: HTML5 Specification – https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5, CSS3 Specification –

[6] NumPyNumPy is a Python library used for numerical computation and array manipulation in image processing operations such as resizing and pixel calculations. Reference: NumPy Documentation –https://numpy.org/doc/ PythonProgrammingLanguage Python serves as the primary backend programming language for integrating MediaPipe, OpenCV, and rembg modules. Reference:PythonOfficialDocumentation–

[7] URL Image Handling (HTTP Requests) The system includes functionality to fetch garments through user-provided URLs using HTTP-based image downloading and image decoding techniques. Reference: Python Requests Library –

[8] Alpha Blending & Image Overlay Techniques Alpha blending is used to merge the transparent garment image with the webcam frame for realistic visual representation. Reference: Gonzalez, R.C., Woods, R.E., Digital Image Processing, Pearson Education, 4th Edition.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)